CHAPTER

17

GETTING STARTED WITH A PRACTICAL AND EFFICIENT TIME-FREQUENCY TOOLBOX TFSAP-7.0[0]

INTRODUCTION AND OVERVIEW

The focus of this chapter is the practical application of the time-frequency (t,f) algorithms described in the book to simulated and real signals, using an advanced flexible platform for Time-Frequency Signal Analysis and Processing (TFSAP). This platform, known as the (t,f) toolbox TFSAP 7.0, provides two options to the user: either to use the available interactive graphical user interface (GUI), or to use online commands in the user's own code. Updates and improvements made to this TFSAP (t,f) toolbox are inline with the latest research done in the (t,f) field. The TFSAP toolbox has proved since its inception to be an efficient and popular tool for analyzing nonstationary signals. Previous versions provided users with an array of tools to analyze nonstationary signals accurately and efficiently [1]. The updated TFSAP (t,f) toolbox extends the functionality of previous versions. It provides additional options such as generating new (t,f) distributions, and synthesizing a signal from a (t,f) image or specifications. Several demo scripts are included in the current version to demonstrate the functionality of the toolbox and to coach new users to use the TFSAP toolbox for advanced signal processing applications dealing with nonstationarities. The version current at the time of publication is renamed TFSAP 7.0 so as to coincide with the second edition of this book. It can be downloaded for free as a service to the community. The toolbox is intended to work in a MATLAB/Octave environment and it includes several databases of real signals, including EEG and other medical data, as an encouragment to foster further research in biomedicine and related fields.

[0]Author: **B. Boashash**, Qatar University, Doha, Qatar; University of Queensland Centre for Clinical Research, Brisbane, QLD, Australia (boualem@qu.edu.qa).

Time-Frequency Signal Analysis and Processing. http://dx.doi.org/10.1016/B978-0-12-398499-9.00017-0

17.1 INTRODUCTION AND (*t*, *f*) TOOLBOX DESCRIPTION

This current updated version TFSAP 7.0 extends and enhances previous TFSA versions continually updated since the first public release in 1987 at the inaugural ISSPA conference in Brisbane, Australia.

In terms of terminology, TFSA refers to the name of the previous versions of the (t,f) toolbox; TFSAP refers to the versions distributed at the time of publication in 2015 and after. Version 7.0 is simply the 1st upgrade made in 2015. This version is provided for free as supplementary material along with several real data sets.

17.1.1 FEATURES AND CONTENTS

17.1.1.1 Standard Original Features of the (t, f) Toolbox

The previous versions of this toolbox[1] broadly addressed the issues of nonstationary signal and noise generation, quadratic time-frequency distributions (QTFDs), multilinear (t,f) analysis, instantaneous frequency (IF) estimation, and tools for data visualization and plotting [1]. For QTFDs, the algorithms included were the Wigner-Ville distribution (WVD), smoothed Wigner-Ville distribution (SWVD), cross-Wigner-Ville distribution (XWVD), spectrogram, Rihaczek-Margenau, exponential distribution (also known as Choi-Williams distribution), and Born-Jordan and Zhao-Atlas-Marks distributions. Multilinear (t,f) analysis is also available with the polynomial WVD of 4th- and 6th-order kernels. In addition, time-scale analysis includes algorithms for wavelet coefficients computation [2]. For IF estimation, a substantial suite of algorithms was also included. These are: the general phase difference, weighted phase difference, adaptive, zero-crossing, least square polynomial fit, peak of spectrogram, peak of WVD [3], and peak of polynomial WVD IF estimators (Section 5.4).

To facilitate the use of TFSA tools and utilities, a GUI was developed in MATLAB with computationally intensive routines written in C for efficient implementation. For MATLAB and C interfacing, MEX file formatting is used. The previous package of TFSA was written to run only on 32-bit Windows or Linux operating system platforms.

A comprehensive user guide and tutorial was also provided as part of the (t,f) toolbox; this is briefly described in the next sections. This tutorial and user guide was designed to give an overview of the features and capabilities of the toolbox as well as provide a step-by-step procedure from installation to usage. The version (TFSAP 7.0) extends this to 64-bit Windows and 64-bit Linux platforms.

17.1.1.2 Features Introduced in the Updated (t, f) Toolbox

Although (t,f)-based signal analysis is a mature research area, a continual improvement is ongoing for the development of more accurate and efficient algorithms/approaches. In particular, an intensive research effort has been done for designing and implementing high-resolution QTFDs as they have proven to be very effective and allowed improvements in terms of resolution and accuracy in the analysis and characterization of nonstationary and/or multicomponent signals. Although the

[1] To the best of the author's knowledge, the core of the earliest (t,f) toolbox was developed by the author during the course of his earlier research in the period 1978-1982, while working at Elf-Aquitaine, France, then upgraded during his stay in INSA, Lyon, France and finally completed it as an interactive Fortran package at the University of Queensland, Australia, in the mid-1990s with the assistance of several PhD students. It was first released publicly as part of the conference ISSPA-1987 in Brisbane, Australia, and then continually upgraded since then as a MATLAB toolbox. Several hundred educational and research institutions obtained this toolbox over the years, with or without source code. (See user manual and supplementary material for more details.)

quadratic class of time-frequency distributions (TFDs) offers good temporal and spectral resolutions simultaneously, these transforms suffer from cross-terms when analyzing multicomponent signals. These cross-terms that result because of the quadratic nature of the transforms may be useful in some applications, such as classification, as they provide extra detail for the recognition algorithm; however, in some other applications, these cross-terms need to be reduced by carefully choosing a smoothing function. This is achieved at the expense of resolution.

These quadratic TFDs were considered in earlier chapters as smoothed versions of the WVD. The choice of a particular TFD depends on the specific application at hand and the properties that are desirable for that application. As a result, many new members of the quadratic class of TFDs have been designed, and the list is still increasing. So there was a need to update the TFSA package. In addition to the standard (t,f) methods listed in the previous section, the new version TFSAP 7.0 includes the following advanced TFDs:

1. B-distribution (BD) and modified B-distribution (MBD) (see Sections 2.7.5 and 5.7 of the book),
2. Extended modified B-distribution (EMBD) (see Section 2.7.5 and Ref. [4]),
3. Compact support kernel TFD (CSK) (see Section 3.3.4 and Ref. [4]),
4. Extended compact support kernel TFD (CKD) (see Section 3.3.4.2),
5. Multidirectional TFD (MDD) (see Sections 3.3.4.3 and 5.9 and Ref. [5]),
6. Enhanced spectrogram (or S-method) (see Sections 2.3.2 and 3.3.4.1)

and several others (see the menu of TFSAP 7.0 for full details).

More importantly, this (t,f) toolbox allows the user to define new data-dependent TFDs suitable for the application considered (see details in Section 17.5).

17.1.2 TEST SIGNALS UPDATE

Researchers use (t,f) methods as an effective tool in many new applications, such as in the fields of biomedical, speech, radar, and sonar signal processing. This presented a greater need to improve the TFSA toolbox to include several test signals with the above-mentioned quadratic TFD-based tools to help researchers perform advanced (t,f)-based analysis on test data relevant to their needs.

More particularly, in the biomedical field, (t,f) methods have shown great potential in a number of applications involving, e.g., electroencephalogram (EEG), heart rate variability (HRV) signals, and fetal movements (FetMov) to mention just a few [4] (see Chapter 16 for details). EEG, HRV, and FetMov abnormalities analysis using TFDs is an important research area, and these datasets are therefore desirable in a toolbox. TFSAP 7.0 contains several such data sets.

Synthesis of a time-domain signal from a given TFD or (t,f) image also has important applications in signal processing. By using the signal-synthesis property, any signal can be reconstructed from TFDs or from a (t,f) image that has a minimum of desirable properties (see Sections 11.2 and 11.5), allowing applications such as denoising, (t,f) filtering, and signal enhancement (see Section 11.6, p.678).

17.1.3 STANDARD FORMULATIONS AND THEIR IMPLEMENTATIONS

A few basic definitions and formulas are reproduced here from earlier chapters for convenience and completeness so that the reader can find in a nutshell what is implemented in the (t,f) toolbox.

Given a real signal $s(t)$, we first form its analytic associate $z(t)$ as follows (see Eq. (1.2.7), p.44):

$$z(t) = s(t) + \mathrm{j}\mathcal{H}\{s(t)\}, \tag{17.1.1}$$

where $\mathcal{H}$ is the Hilbert transform (see Section 1.2.3). We then define the QTFD as per Eq. (3.2.13), p.113, i.e., as:

$$\rho_z(t,f) = W_z(t,f) \underset{tf}{**} \gamma(t,f), \tag{17.1.2}$$

where $\gamma(t,f)$ is a 2-D smoothing kernel and $W_z(t,f)$ is the WVD defined in Chapter 2 as (see Eq. (2.1.17), p.69):

$$W_z(t,f) = \int_{-\infty}^{\infty} z(t+\frac{\tau}{2})z^*(t-\frac{\tau}{2})e^{-\mathrm{j}2\pi\tau f}\,\mathrm{d}\tau = \underset{\tau\to f}{\mathcal{F}}\{z(t+\frac{\tau}{2})z^*(t-\frac{\tau}{2})\}. \tag{17.1.3}$$

The equivalent discrete-time QTFD of Eq. (17.1.2) is given in Section 6.1 and Refs. [4,6] as:

$$\rho[n,k] = W[n,k] \underset{n}{\circledast} \underset{k}{\circledast} \gamma[n,k]|_{k=0,1,\ldots,N-1} \tag{17.1.4}$$

with

$$W[n,k] = \sum_{m=0}^{2N-1} K[n,m]e^{-\mathrm{j}2\pi mk/2N} = \underset{m\to k}{\mathrm{DFT}}\{K[n,m]\}, \tag{17.1.5}$$

where $\circledast$ is the circular convolution; N, n, k, and m are the signal length, discrete time, discrete frequency, and discrete lag, respectively; $W[n,k]$ is the discrete WVD, and $K[n,m]$ represents a discrete-time version of $K(t,\tau) = z(t+\tau/2)z^*(t-\tau/2)$ as detailed in Chapters 2 and 6.

17.2 TECHNICAL CONTENT OF THE (*t*, *f*) TOOLBOX TFSAP 7.0

17.2.1 LIST OF UPDATED FUNCTIONS IN VERSION 7.0

In order to update the features of the previous versions of TFSA toolbox, the following options are included in the 2015 version renamed TFSAP 7.0[1]:

1. **Generation of test signals and noise such as:** linear frequency modulation (FM), quadratic FM, cubic FM, stepped FM, sinusoidal FM, hyperbolic FM and Gaussian and uniform noise; others to appear in future versions.
2. **Generation of demo signals (signal1, whale1, bat1, eeg1, hrv1):** signal1 is an arbitrary test signal; whale1 is a whale signal; bat1 is a bat signal; eeg1 is an EEG signal; and hrv1 is an HRV signal. Other signals such as FetMov, bird, and speech are also included in this and future upgrades.
3. **Generation of TFDs such as:** WVD, SWVD, spectrogram, Rihaczek-Margenau-Hill distribution, Windowed-Rihaczek-Margenau-Hill distribution, exponential distribution, B-distribution, MBD, EMBD, CSK-based distributions, extended CSK-based distributions, Born-Jordan distribution, Zhao-Atlas-Marks distribution, XWVD, polynomial Wigner-Ville distribution (order-6 kernel), polynomial Wigner-Ville distribution (order-4 kernel), and ambiguity function (AF); S-method; MDD; others to appear in future upgrades.

[1] Next updates will be uploaded on a regular basis on several websites, including the author's webpage, Elsevier book website, The University of Queensland espace, Qatar University Qspace, and other relevant websites associated with the author.

4. **Generation of time-scale signal representations:** Wavelet transforms (WTs) and scalogram; others to be added in future upgrades.
5. **IF estimation algorithms:** Finite-phase difference including: first-order (FFD), second-order (CFD), fourth-order and sixth-order, weighted-phase difference, zero crossing, adaptive least mean square (LMS), adaptive Recursive Least Squares, least squares polynomial coefficients, peak of the spectrogram, peak of the WVD, and peak of the polynomial WVD; other options to be added in future updates.
6. **Signal synthesis using the following TFDs:** Short-time fourier transform (STFT), spectrogram, and WVD.
7. **Other digital signal processing (DSP) tools:** Analytic signal calculation, power spectrum, and previously mentioned demo signals (e.g., signal1, whale1, bat1, eeg1 and hrv1).
8. **Data visualization routines including:** plot, image, pseudocolor, waterfall, mesh, surf, contour, and tfsapl.
9. **The current version of TFSAP toolbox runs on the following four different operating system platforms:** 32-bit Windows, 64-bit Windows, 32-bit Linux, and 64-bit Linux; others to appear in future upgrades.

17.2.2 TFSAP DEVELOPMENT PROCESS

TFSAP has been developed in MATLAB and C as an extension of a previous Fortran and C toolbox that were themselves an extension of an APL version.[2] The GUI of TFSAP is developed in MATLAB, while the computationally intensive scripts are written in C for efficient implementation. The MATLAB and C code communicate with each other using MEX file formatting.

Adding a new function to TFSAP requires the coding of source files in C to perform the required function. Then TFSAP GUI files are edited to provide access to the new function. For example, to add the EMBD in TFSAP requires editing the source code of the quadratic function in C, as the EMBD is a member of the quadratic class of TFDs. Afterward, the source code of the quadratic TFD analysis GUI is updated to incorporate the functionality of the EMBD in the GUI of the quadratic (t,f) analysis. This task is performed by the (t,f) toolbox maintainers.

17.3 USER'S GUIDE FOR (*t, f*) TOOLBOX TFSAP-7.0

This section takes a step-by-step approach to the understanding and use of this (t,f) toolbox. It also has an adjunct purpose in that it familiarizes the user with TFSAP 7.0 for MATLAB. TFSAP 7.0 has been written to run on a platform supporting MATLAB. A version supporting OCTAVE is under development, and it is planned to be part of the next version, TFSAP 7.1.

17.3.1 INSTALLATION

The MATLAB (t,f) toolbox TFSAP 7.0 can be downloaded from several links, including www.time-frequency.net, the Elsevier book website, http://espace.library.uq.edu.au/view/uq:211321, and the author's personal page. The corresponding code is described in Ref. [9] and Chapter 6.

[2]The main code of the APL version appears in the Appendix in the 1982 PhD thesis of the author. The first working version was published by the author in an ELF-Aquitaine company report dated September 1978 [7] and reported in Ref. [8].

The versions for Linux and MS-Windows are distributed pre-compiled. To install them, just extract the zipped file and follow the instructions in the README text file. To run TFSAP 7.0, you need to have MATLAB already installed. To run the TFSAP 7.0 GUI, a minimum system requirement of 32 megabytes RAM on PCs is needed to avoid memory problems.

17.3.2 RUNNING *TFSAP* 7.0

To run TFSAP 7.0 on MS-Windows, all you need to do is to click the MATLAB icon on the screen and then type: `tfsa7`
at the MATLAB prompt.[1] If you are running TFSAP 7.0 on UNIX workstations, invoke MATLAB from the shell by typing: `matlab`

(followed by the "Enter" key), and then type: `tfsa7`

at the MATLAB prompt. The Main Menu of the GUI will then appear as shown in Fig. 17.3.1. The GUI of TFSAP 7.0 contains the pop-up menus and interface fields which control the various functions and tools available in the toolbox. The GUI provides an alternative to typing TFSAP 7.0 commands on the MATLAB command line, to make it easy for beginners.

It contains four buttons. These are:

- Welcome: Clicking this button gives general information on TFSAP-7.0.
- Copyright: The license agreement screen appears by clicking this button.

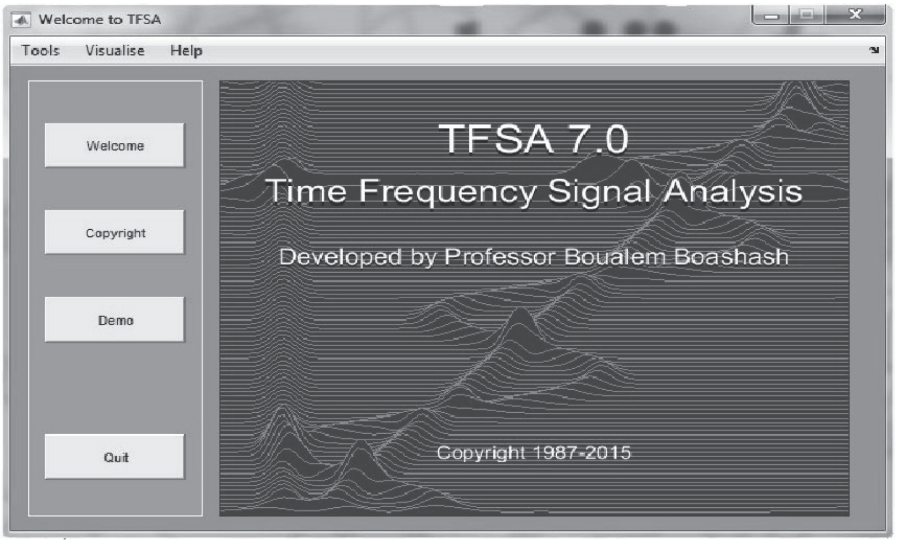

FIGURE 17.3.1

Main menu of GUI of TFSAP 7.0 toolbox

[1] If you use an earlier version, e.g., TFSA 5.5 or 6.0, then the number seven changes to, e.g., five for version 5.5 or six for version 6.0 (in this case, consult the manual accompanying the software).

- Demo: Clicking this button gives a demonstration of TFSAP-7.0.
- Quit: Clicking this button aborts the TFSAP session.

The TFSAP 7.0 menu bar is composed of three major menu items: Tools, Visualize, and Help. Any menu item can be opened by placing the mouse's pointer on the corresponding item and then clicking the primary button of the mouse. You can access any menu sub-item by clicking with the primary button of the mouse on the corresponding menu item and while holding this button down, dragging the mouse across the submenu bar until encountering the required subitem, then releasing the mouse button.

17.3.3 DATA ACCESSES AND SIGNAL GENERATION GUI

Figure 17.3.2 shows the main GUI through which different types of nonstationary signals can be generated. Figure 17.3.2 shows an example of generating a linear FM signal with 128 samples, starting and ending frequencies of 0.1 and 0.4 Hz. There is also an option to generate the real or analytic form of a test signal. The complex analytic signal (see Chapter 1) can be generated by adding to the time-domain signal an imaginary part using the Hilbert transform. In TFSAP, the analytic signal is generated using an equivalent frequency domain method (see Chapters 1 and 6).

The different types of test signals that can be generated from TFSAP 7.0 are listed in Section 17.2.1. There is also an option to generate noise only or a test signal with noise. An illustration to generate a quadratic FM noisy signal is shown in Fig. 17.3.3(a). A Gaussian noise of signal-to-noise ratio (SNR) 20 dB is added with the signal. The length of the signal is 128 samples with starting frequency of 0.1 Hz and ending frequency of 0.4 Hz. As previously mentioned, TFSAP 7.0 also includes several real-world nonstationary signals. These can be generated by selecting Demo Signal as signal type. Then, examples of demo signals will appear in the Demo Signal tab as shown in Fig. 17.3.3(b).

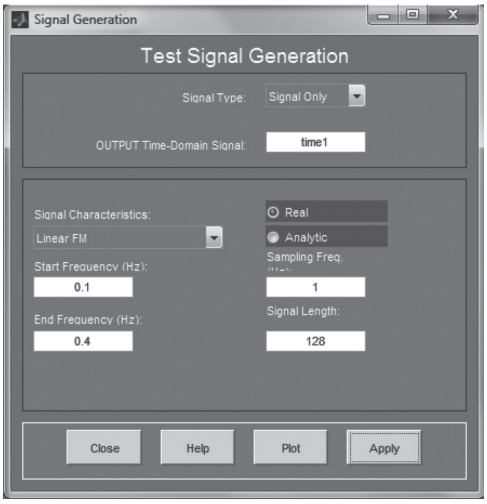

FIGURE 17.3.2

Signal generation GUI.

FIGURE 17.3.3

Generating test signals: (a) a quadratic FM signal with noise; (b) a demo signal.

All references to "signal" or "array" names in TFSAP 7.0 refer to variables in the MATLAB workspace. Values are read from and written to this space, and can be accessed normally from the MATLAB command line.

Thus two signals can be added together using the following procedure:

1. Generate time1 from the GUI Signal Generation popup.
2. Generate time2 from the GUI Signal Generation popup.
3. In the MATLAB command-line window, type: `time3 = time1 + time2`
4. Then, time3, the sum of the two signals, appears in the workspace and it can now be accessed from the GUI.

17.3.4 IF ESTIMATION GUI

As described in the previous chapters, IF estimation is an important technique in (t,f) analysis (see Chapter 1 and Ref. [3]). There are several algorithms that can be used to compute IF, e.g., differentiation schemes, adaptive techniques, peaks of TDS, and zero-crossings (for more details, see Chapter 10 and Ref. [10]). The detailed list of functions for IF estimation is given in Section 17.2.1. For any signal, the user needs to perform the following tasks for IF estimation:

- Generate a time-domain signal as explained in Section 17.3.3.
- Open the IF estimation GUI.
- Enter the name of the signal in the Input Time-Domain Signal textbox

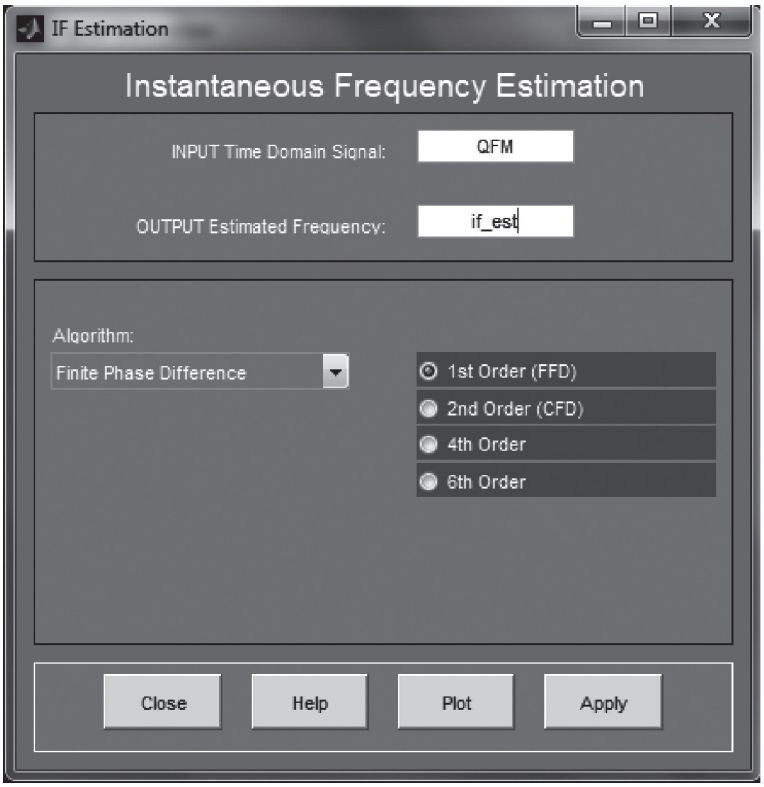

FIGURE 17.3.4

GUI for IF estimation.

- Enter the name of the output in the Output Estimated Frequency textbox
- Select any IF estimation method from the list of methods shown in the algorithm tab and then click the Apply button to generate the IF.

Figure 17.3.4 shows an example of the IF estimation GUI for a quadratic FM signal.

17.3.5 QUADRATIC TFD ANALYSIS GUI

17.3.5.1 Using an Existing Time-Frequency Method

In TFSAP 7.0, a GUI is included that gives a menu with the option to generate different TFDs for any signal. For analyzing the quadratic TFD of any signal, the user needs to perform the following tasks:

- Generate a time-domain signal as explained in Section 17.3.3.
- Open the quadratic TF analysis GUI.
- Enter the name of the signal in the Input Time Domain Signal textbox
- Enter the name of the output in the Output TFD textbox

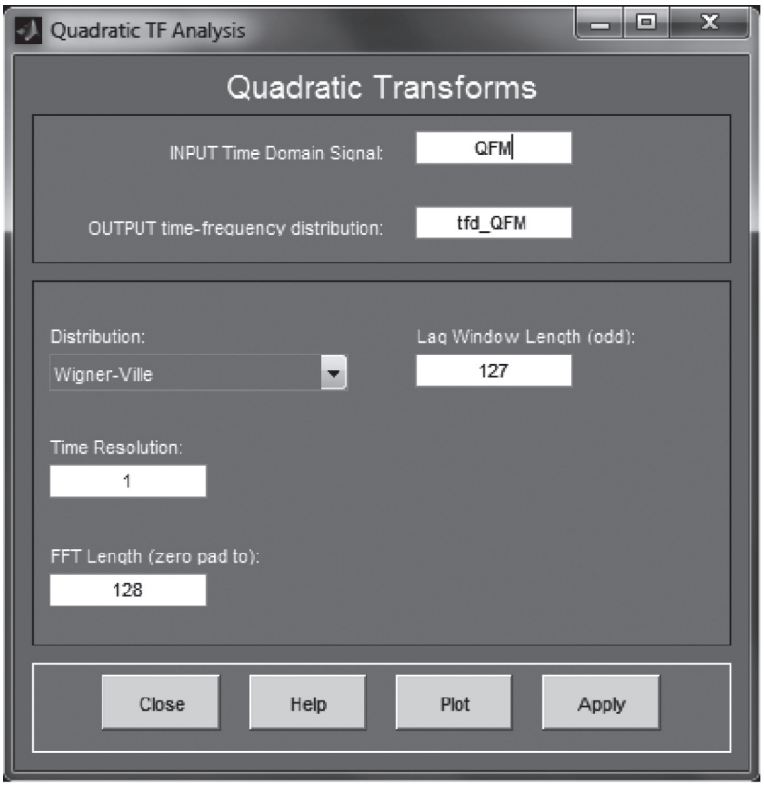

FIGURE 17.3.5

GUI for quadratic TF analysis.

- Select any quadratic TFD from the list shown in the distribution tab
- Set the FFT length and lag window length and press apply to generate quadratic TFD (see Chapter 6 for more details).

Figure 17.3.5 shows an example of estimating the WVD of a quadratic FM signal using the quadratic (t,f) analysis GUI.

17.3.5.2 User-Defined TFD

If the user wants to define and test a new kernel, he can do so following the procedure described below:

1. Define the kernel $\gamma[n,k]$ in the MATLAB line command as per Eq. (17.1.4).
2. Generate the WVD $W_z[n,k]$ of the signal z as described above.
3. Perform a double discrete convolution product between $W_z[n,k]$ and $\gamma[n,k]$ (in n and k) using the MATLAB command `conv2` (see Eq. (17.1.2)). Then, the result of the above operation is the TFD obtained using the kernel $\gamma[n,k]$ defined in step 1.
4. Proceed to plot the above result or perform any other desired operation.

   For more details, see Sections 17.1.3 and 17.5.2.

17.3.6 SIGNAL SYNTHESIS GUI

A time-domain signal $z[n]$ can be synthesized from a given TFD $\rho_z[n,k]$ under some conditions (see Sections 2.3.1, 3.1.1, and 11.2) or from any (t,f) image. TFSAP 7.0 includes a GUI to help users explore and use the different features of signal synthesis in applications such as (t,f) filtering and signal enhancement. The main GUI for signal synthesis is shown in Fig. 17.3.6. To reconstruct a discrete time-domain signal $z[n]$ from a given discrete (t,f) image $\rho_z[n,k]$, the user needs to perform the following tasks:

- Generate a desired (t,f) image (e.g., the modified TFD of a particular signal).
- Open the signal synthesis GUI.
- Enter the name of the discrete (t,f) image in the Input TFD textbox.
- Enter the name of the output in the Output time domain signal textbox.
- Select any synthesis method along with the lag window length and then click Apply to generate a time-domain signal.

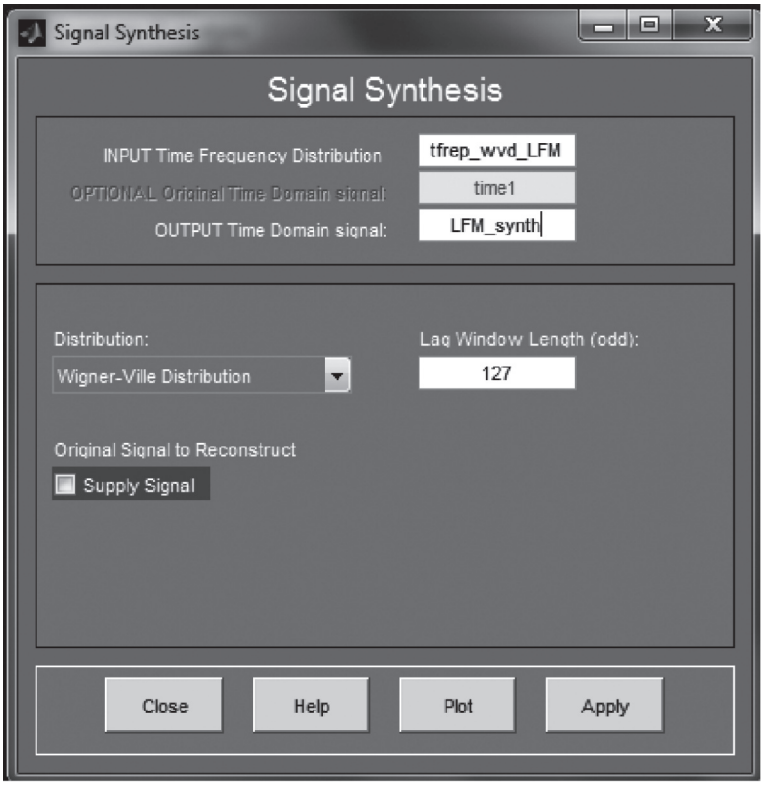

FIGURE 17.3.6

GUI for signal synthesis (note that the input here is similar to the output of the quadratic transform shown in Fig. 17.3.5).

17.3.7 TFSAP-MATLAB FLEXIBLE INTERFACE

17.3.7.1 Compatibility with Other MATLAB Functions

The GUI is an interface to the TFSAP 7.0 functions. All normal MATLAB features remain in place. For example, titles, labels, figure properties and variables may all be changed using the MATLAB command-line interface. Figures can be printed using the MATLAB's print command, as normal.

17.3.7.2 MATLAB Command-Line Interface

TFSAP 7.0 functions can be accessed in the same manner as the standard MATLAB functions (see Sections 17.4.3.3, 17.4.4, and 17.5). For a list of functions in TFSAP 7.0 type: `help tfsa7`

Help on any of the functions can be obtained by typing: `help` ⟨*functionname*⟩. A detailed list of functions that TFSAP 7.0 supports is given in Section 17.2.1.

17.4 A BRIEF (*t*, *f*) TUTORIAL USING THE (*t*, *f*) TOOLBOX

This section illustrates the usability of the updated TFSAP 7.0 toolbox for real-life applications. It is a concise version of a tutorial that is part of the (t,f) toolbox.

17.4.1 RESULTS FOR SIMULATED SIGNALS

17.4.1.1 Results for a Quadratic FM Signal

A comparison of TFDs for a quadratic FM signal is shown in Fig. 17.4.1. The frequency of quadratic FM signal (length = 128 samples) ranges from 0.1 to 0.4 Hz. Figure 17.4.1(a) shows the WVD; Fig. 17.4.1(b) shows the exponential distribution with sigma 20; Fig. 17.4.1(c) shows the spectrogram with a smooth "hamm" window and window length of 15 samples; and Fig. 17.4.1(d-f) shows the B-distribution with $\beta = 0.1$ for a quadratic FM signal, then the MBD, followed by the EMBD. All the TFDs are generated with lag window length of 15 samples, time-resolution of 1, and length of FFT 128 samples.

17.4.1.2 Results for IF Estimation of Quadratic FM Signals

Figure 17.4.2 illustrates the IF estimation of a Quadratic FM signal using three different methods; i.e., zero-crossing, phase differentiation, and peaks of WVD, all available from TFSAP 7.0. For more advanced high-resolution TFDs, such as the MBD or EMBD, the peak extraction method can yield an IF estimation with different properties. The user is encouraged to perform this comparison.

17.4.2 RESULTS FOR REAL SIGNALS

17.4.2.1 Results for a Bat Signal

Several real demo signals are included in the toolbox, i.e., bat signal, whale signal, EEG and HRV signals. A comparison using two different quadratic TFDs on a bat signal sampled at 142 kHz is shown in Fig. 17.4.3. Figure 17.4.3(a) shows the WVD of the bat signal with a lag window length of 63 samples and a time-resolution of 3. Figure 17.4.3(b) shows the modified B-distribution of the bat signal with a lag window length of 63 samples, a time-resolution of 3, and $\beta = 0.01$. From Fig. 17.4.3, it can be observed that the bat signal is multicomponent with a decreasing nonlinear FM. It can also be observed that the MBD performs much better in suppressing the cross-terms. (Note that increasing the value of the time resolution results in a less detailed figure, but with improved visual clarity as the

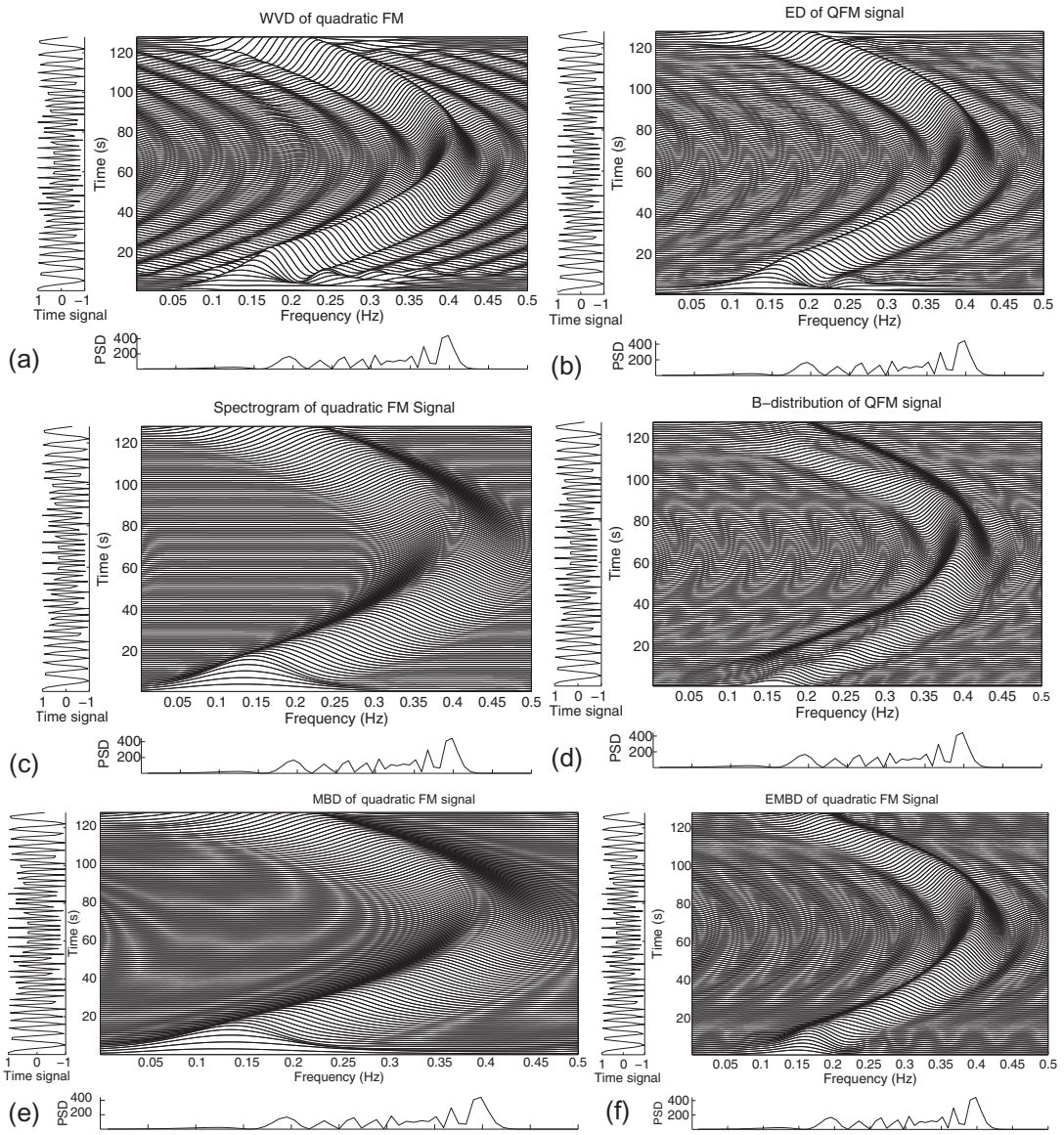

FIGURE 17.4.1

Comparison of standard quadratic TFDs for a quadratic FM signal: (a) Wigner-Ville distribution; (b) exponential distribution; (c) spectrogram; (d) B-distribution; (e) MBD; (f) EMBD.

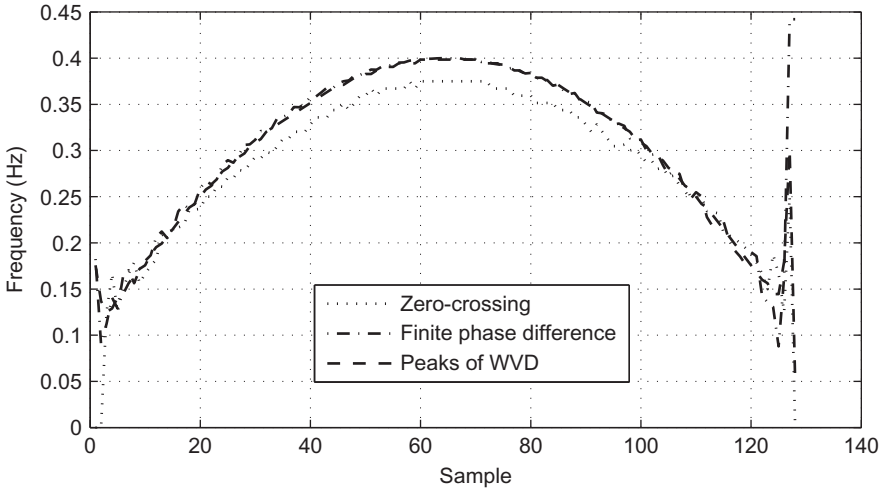

FIGURE 17.4.2

IF estimation of a quadratic FM signal. (Note that the results of the phase differentiation and peaks of WVD methods are very close in this example.)

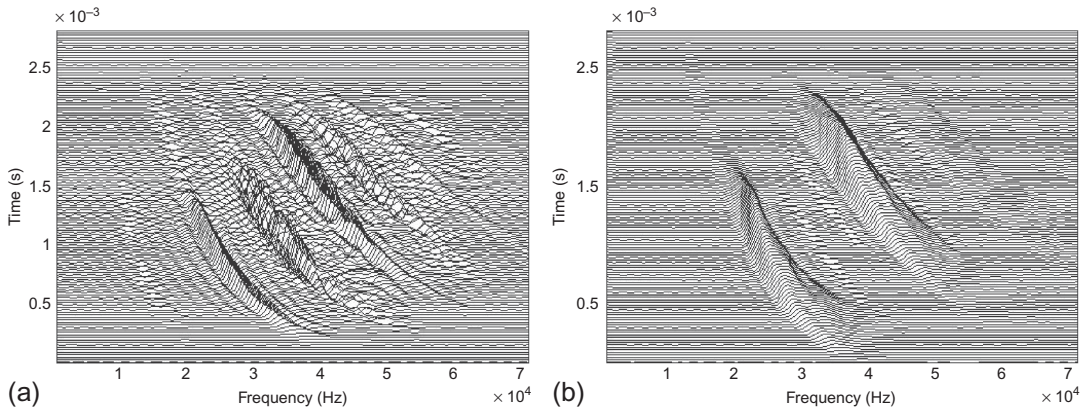

FIGURE 17.4.3

Comparison of WVD and MBD for bat echolocation signal: (a) WVD of a bat signal; (b) MBD of a bat signal.

number of time lines decreases. This is demonstrated in Fig. 17.4.4 where a time resolution of 1 was used in Fig. 17.4.4(a) and a time resolution of 3 was used in Fig. 17.4.4(b) for the WVD of a linear FM signal.)

17.4.2.2 Results for an EEG Signal

EEG signal analysis based on quadratic TFDs is illustrated in Fig. 17.4.5. Graph (a) shows the EMBD of an EEG signal. Figure 17.4.5 shows that the EMBD enhances the ridge structure of multicomponent EEG signal, and it significantly reduces the cross-terms [4].

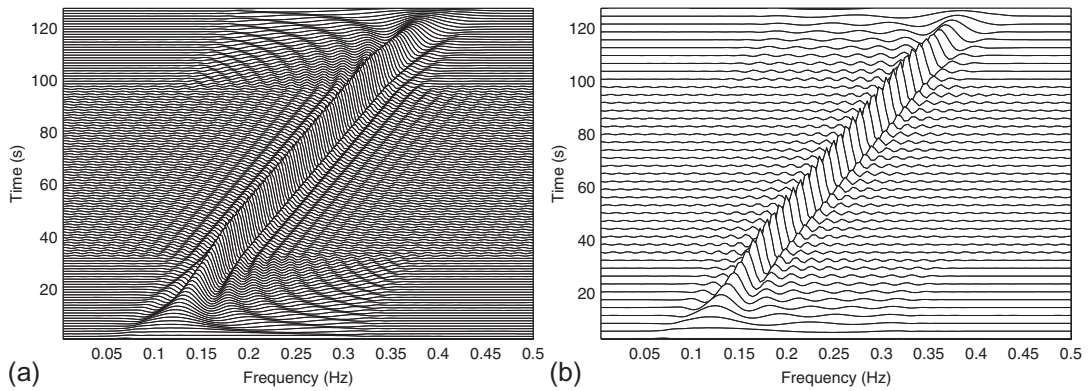

FIGURE 17.4.4

WVD of a linear FM signal: (a) time resolution = 1; (b) time resolution = 3.

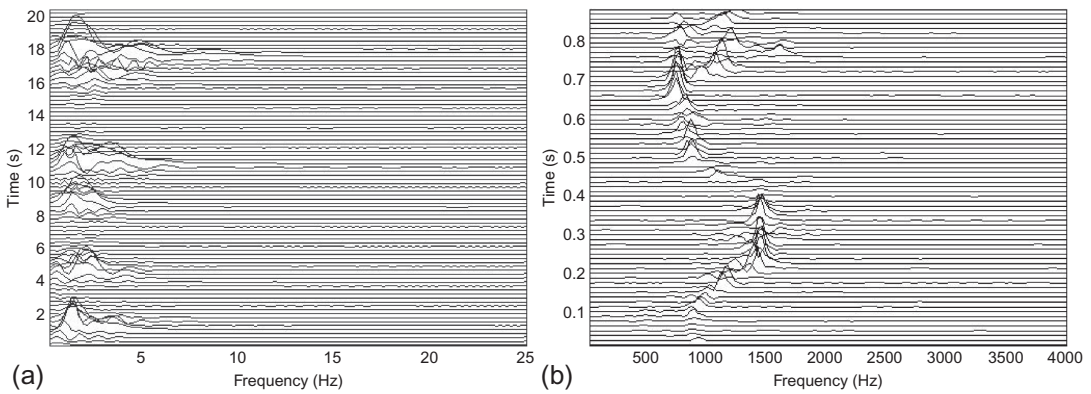

FIGURE 17.4.5

Time-frequency representations of real signals: (a) EMBD of EEG signal; (b) EMBD of signal whale1.

17.4.2.3 Result for a Whale Signal

Several whale signals are also included in the TFSAP 7.0 package. The signal whale1 contains 7000 data points and was collected at a sampling rate of 8 kHz. Figure 17.4.5(b) shows the EMBD of the signal whale1. A second whale signal, whale2, presented in Chapter $\mathcal{I}$, is also included in this toolbox (see Figure $\mathcal{I}$.1.1).

17.4.2.4 Result for an HRV Signal

HRV signals are formed from electrocardiogram (ECG) signals [4]. An HRV signal with length of 256 samples and sampling frequency 2 Hz is included in the TFSAP 7.0 package. Figure 17.4.6(a) shows the MBD of the HRV signal using the tfsapl function, and Fig. 17.4.6(b) shows the same TFD as in Fig. 17.4.6(b) by using the surf plotting function.

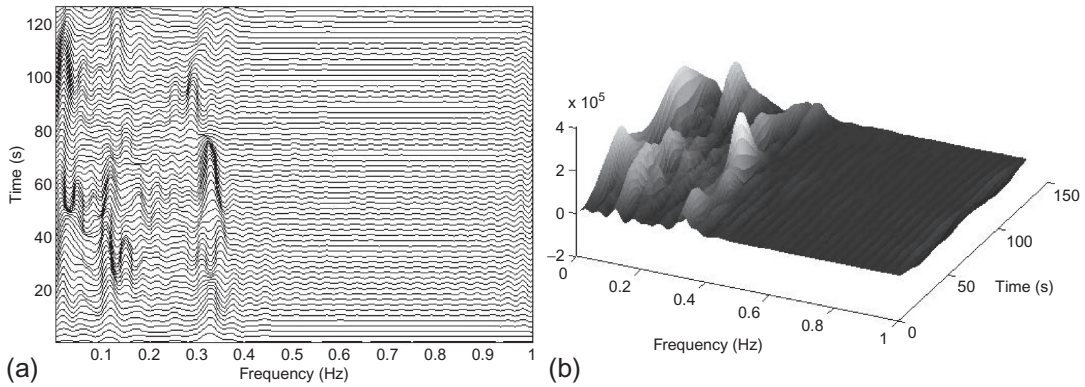

FIGURE 17.4.6

MBD of HRV epoch from a healthy adult: (a) graph obtained using tfsapl function; (b) graph is plotted using surf function with the same MBD parameters as (a): $\beta = 0.02$ and the lag window length is 63. From the figure, three components and their variations can be observed.

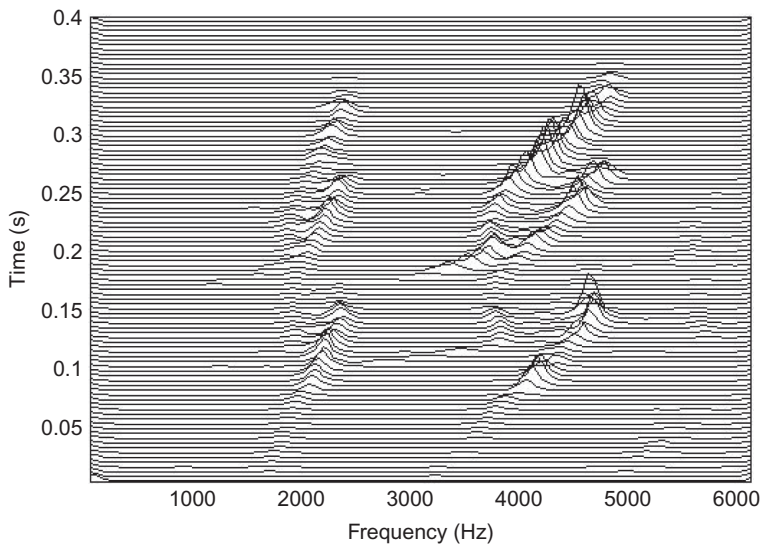

FIGURE 17.4.7

EMBD of a bird signal.

17.4.2.5 Result for a Bird Signal

This signal is a bird song recorded in Australia; it is shown on the book's cover of the first edition. The signal is sampled at 24417 Hz and is 9767 samples long. This signal, called bird1, is included in TFSAP 7.0. Figure 17.4.7 shows the EMBD of the bird signal, obtained with $\alpha = 0.05$, $\beta = 0.5$, a lag window length of 127 samples, a FFT length of 128 samples, and a time resolution of 50. The signal is

downsampled by a factor 2 because after 6 kHz, the signal energy is insignificant. Section 6.6 shows a TFD of the same signal using a separable Hamming/Hanning kernel.

17.4.3 RESULTS FOR SIGNAL SYNTHESIS AND (*t, f*) FILTERING

17.4.3.1 Signal Reconstruction and Denoising Using the WVD

17.4.3.1.1 Signal reconstruction

An example of signal synthesis using the WVD is shown in Fig. 17.4.8: (a) shows a linear FM signal with starting and ending frequency of 0.1 and 0.4 Hz, respectively with signal length of 127 samples; (b) shows the corresponding WVD of the signal; and (c) shows the synthesized time-domain

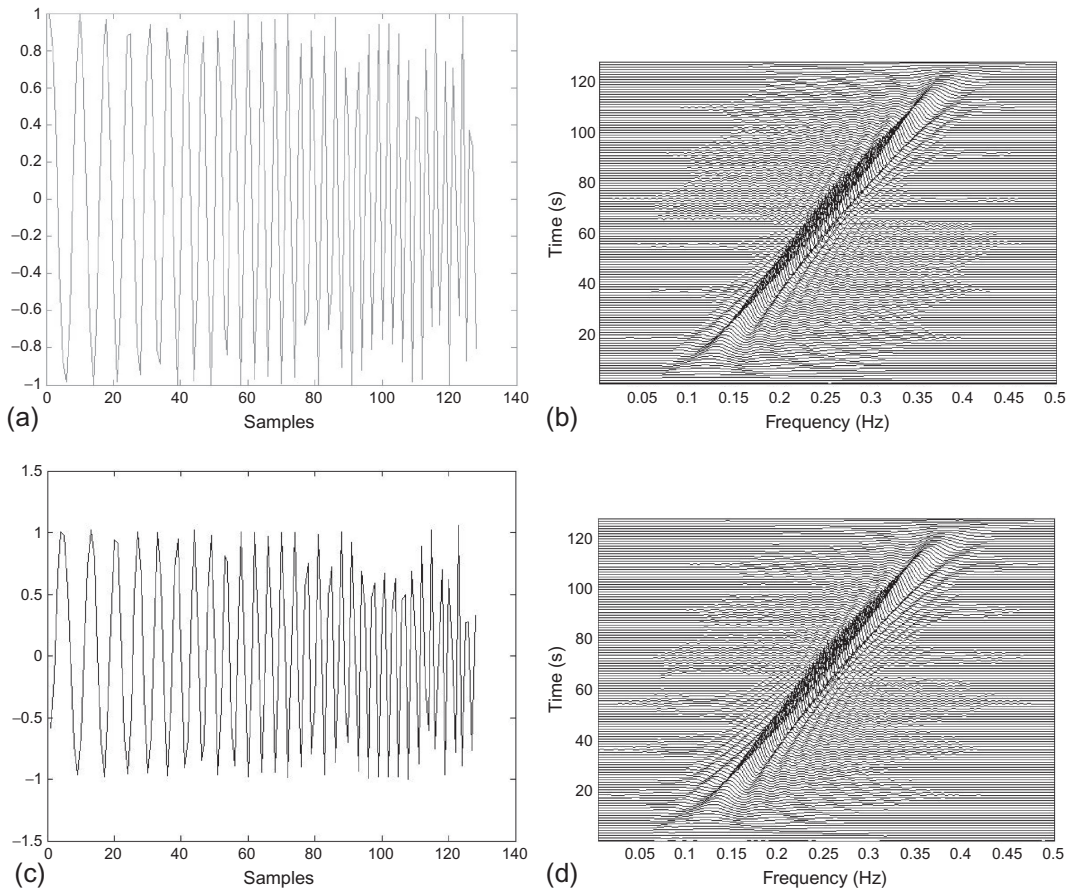

FIGURE 17.4.8

Example of signal reconstruction using the WVD method: (a) signal; (b) WVD of signal; (c) synthesized signal; (d) WVD of synthesized signal.

signal using the WVD method. Graph (d) shows the WVD of synthesized time-domain signal. The results indicate that the reconstructed and original are similar.

17.4.3.1.2 Denoising

Figure 17.4.9 shows the (t,f) filtering of a noisy FM signal with starting and ending frequency of 0.1 and 0.4 Hz, respectively and having a Gaussian noise of 20 dB. This is followed by a signal reconstruction stage. In this example, the following steps are performed to remove noise from this signal:

1. The IF of the signal is estimated by detecting peak locations in the WVD of the noisy signal.
2. The IF estimates are then used to create a binary image such that the value is one along the IF points and zero elsewhere.
3. The binary image is smoothed by convolving it with a 1D Gaussian filter.
4. The smoothed image is then used to mask the WVD.
5. The masked WVD is then "inversed" to yield a reconstructed (t,f) filtered time-domain signal.

Figure 17.4.9(a) shows the time domain representation of the noisy signal. Figure 17.4.9(b) shows the WVD of the noisy signal. Figure 17.4.9(c) shows the filtered TFD of the noisy signal. Figure 17.4.9(d) shows the time-domain representation of the filtered TFD; it shows that (t,f) filtering has significantly reduced the effect of noise and the reconstructed signal closely matches the original one as shown in Fig. 17.4.8(a). In order to show the effect of filtering, the WVD of a synthesized filtered signal is again computed as shown in Fig. 17.4.9(e). By comparing the results of Fig. 17.4.9(b) with Fig. 17.4.9(e), it can be observed that the noise of the synthesized signal is significantly reduced and the WVD of the synthesized signal is closer to the actual WVD of the original signal.

17.4.3.2 Using the STFT and Spectrogram

Section 2.3.1 shows that a signal can be reconstructed exactly from its STFT. The following steps may be performed to remove noise from a signal using the STFT:

1. compute the STFT of the time-domain signal,
2. apply a mask on the STFT (element-by-element multiplication between the STFT coefficients and the mask),
3. the masked STFT is "inversed" to reconstruct a filtered time-domain signal.

Sections 11.2 and 11.6 show several examples of (t,f) signal denoising using the STFT. In particular, Section 11.2 describes in detail how to efficiently implement the inverse STFT, taking into account issues such as window overlap.

17.4.3.3 Time-Frequency Synthesis of a Signal from an Arbitrary Image

Let us present an example of signal synthesis from an arbitrary image describing some desired (t,f) specifications. The procedure involves the following steps:

1. Form the desired (t,f) image.
2. Import the image in the MATLAB environment, i.e.,

```
tfd = double(imread('signal1.jpg','jpg'));
```

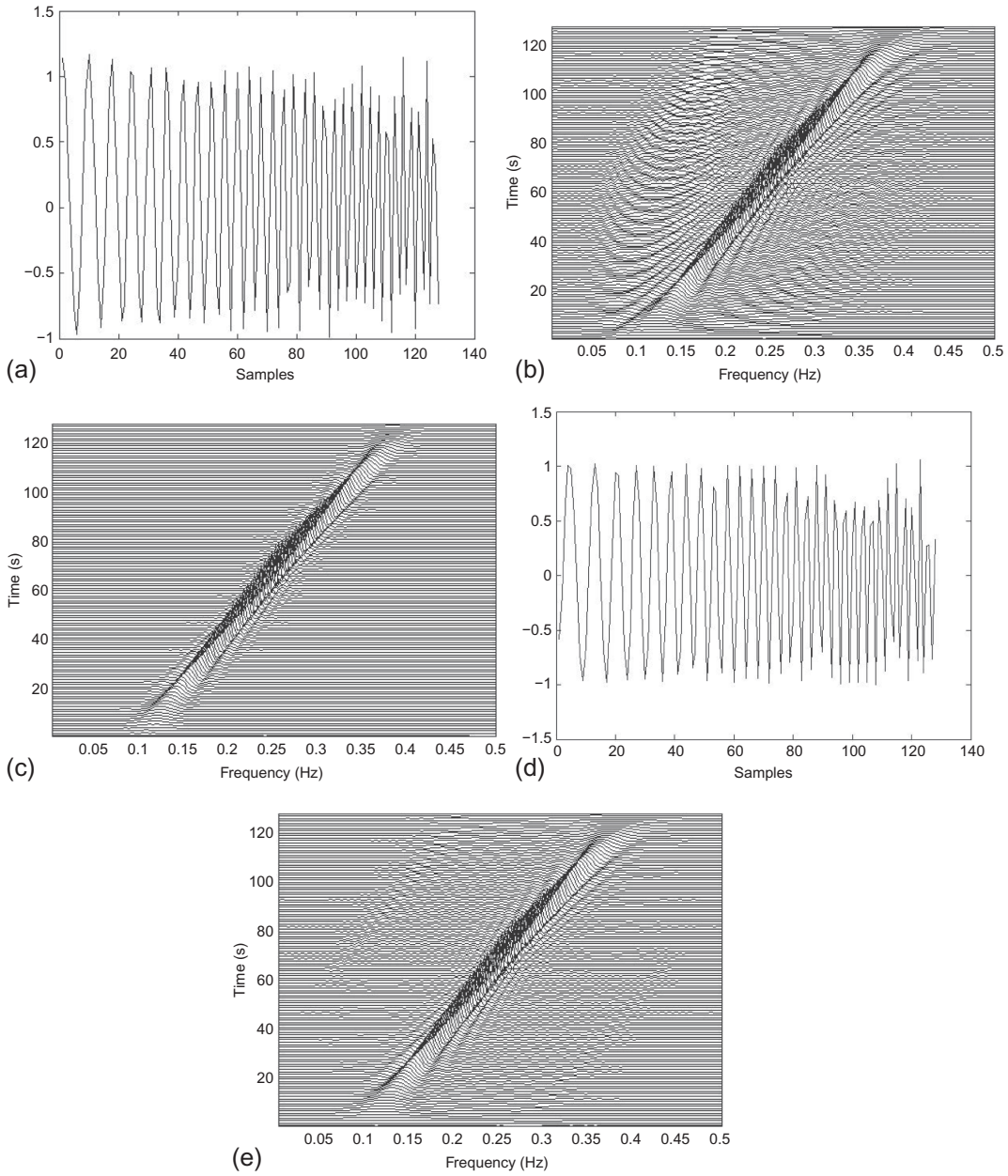

FIGURE 17.4.9

Signal synthesis of a noisy signal using WVD-based TF filtering: (a) signal + noise; (b) WVD of noisy signal; (c) filtered TFD; (d) synthesized signal; (e) WVD of synthesized signal.

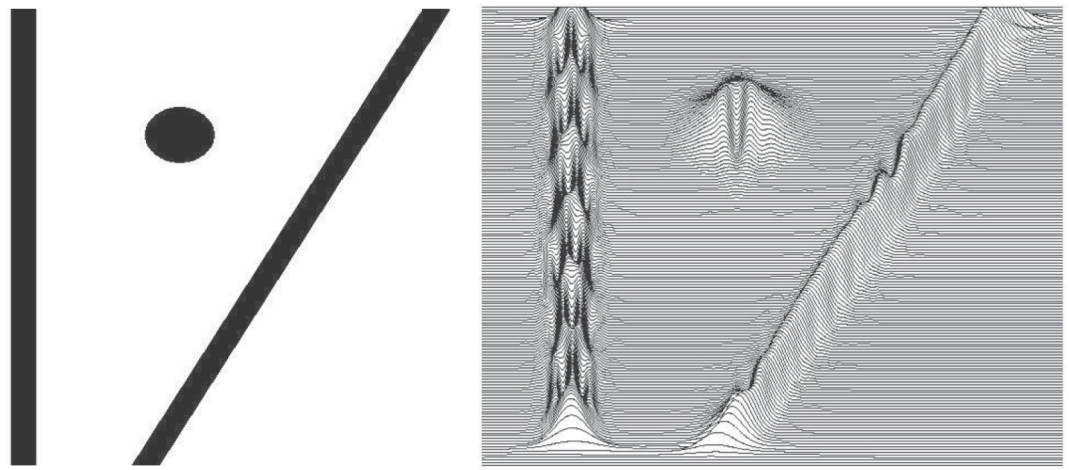

FIGURE 17.4.10

Illustration of signal synthesis from an image: (a) image to synthesize; (b) spectrogram of the reconstructed time-domain signal.

3. Transform the image in gray level (note that this step is necessary only for rgb images):
   `tfd = rgb2gray(tfd);`
4. Transpose the image:
   `tfd = tfd.';`
5. Follow the GUI procedure described in Section 17.3.6; e.g., choose the "Modified Spectrogram" technique.
6. Use the following parameters: "Smoothing Window Length"=127 and "Iterative Tolerance Level"=0.45.

Figure 17.4.10 shows the desired (t,f) image used in this example and the spectrogram of the synthesized time-domain signal. The parameters used to generate the TFD are: a window length of 127 samples, a time resolution of 3, a Hamming smoothing window, and a 512 FFT length.

17.4.4 DUAL USE OF (t, f) TOOLBOX (GUI OR COMMAND LINE)

The toolbox can be used in two different ways: (1) using the GUI as described before; or (2) using the command line. TFSAP 7.0 functions can be accessed from MATLAB in the same manner as the standard MATLAB functions. For a list of functions in TFSAP 7.0, type:

`help <functionname>.`

Further information can be found typing:

`type <functionname>.`

The next section provides more details and an example.

17.5 EXPANDING AND PERSONALIZING THE *TFSAP* TOOLBOX

17.5.1 PRINCIPLE AND ILLUSTRATION

Using the MATLAB command line, it is possible to use most TFSAP functions in one's personal code. For example, let us present how one can use the user-defined kernel option defined in Section 17.3.5.2. In this example, we use a separable Hamming/Hanning kernel (see Sections 3.2.5.1 and 5.7):

1. Generate a signal z, e.g., a quadratic FM signal in the range 0.1 and 0.4 Hz:
   ```
   z = gsig('quad',0.1,0.4,128,1);
   ```
2. Define the kernel $\gamma[n,k]$; in this example, we use an average filter:
   ```
   gamma = hann(11)*hamming(23)';
   ```
3. Generate $W_z[n,k]$ of 128 point signal using a time resolution of 3:
   ```
   tf = wvd(z, 127, 3);
   ```
4. Calculate the double convolution product between $W_z[n,k]$ and $\gamma[n,k]$ to get the TFD associated to the user-defined kernel:
   ```
   tfnew = conv2(tf,gamma,'same')
   ```
5. Plot the TFD using the tfsapl function:
   ```
   tfsapl(z, tf);
   ```

A list of all these command lines are available in the reference guide provided in the (t,f) toolbox (available as supplementary materials).

17.5.2 EXERCISE

Redo the same procedure presented above for the EMBD; define the kernel $\gamma[n,k]$ corresponding to this TFD.

17.6 SUMMARY AND CONCLUSIONS

This current TFSAP toolbox 7.0 provides an excellent opportunity for students and researchers to start their journey in the field of nonstationary signal analysis. It allows the users to exploit the wide range of (t,f) tools in different signal processing application. The version TFSAP 7.0 updates and extends the functionality of previous versions of the TFSA toolbox. The author intends to continue to update it with his team in order to reflect new ideas and algorithms such as Ref. [11], and to improve functionality. This version TFSAP 7.0 is intended also to be available in Octave and to include more biomedical data for testing and research. It should become available on "www.time-frequency.net" the author's personal webpage, on the website of this book (under supplementary material), on The University of Queensland espace, on Qatar University Qspace, and on other associated websites.

REFERENCES

[1] B. Boashash, "Efficient software implementation for the upgrade of a time-frequency signal analysis package", in: Proc. Eighth Australian Joint Artificial Intelligence Conference (A.I.'95), 1995, pp. 26-31.

[2] I. Daubechies, "The wavelet transform: a method for time-frequency localization", in: S. Haykin (Ed.), Advances in Spectrum Analysis and Array Processing, vol. 1, Prentice-Hall, Englewood Cliffs, NJ, 1991, pp. 366-417 (Chapter 8).
[3] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal—Part 1: Fundamentals", *Proc. IEEE* 80 (4) (1992) 520-538.
[4] B. Boashash, G. Azemi, J.M. O'Toole, "Time-frequency processing of nonstationary signals: advanced TFD design to aid diagnosis with highlights from medical applications", *IEEE Signal Process. Mag.* 30 (6) (2013) 108-119.
[5] B. Boashash, N.A. Khan, T. Ben-Jabeur, "Time-frequency features for pattern recognition using high resolution TFDs: a tutorial review", *Digit. Signal Process.* 40 (2015) 1-30, http://dx.doi.org/10.1016/j.dsp.2014.12.015.
[6] A.P. Reilly, G.J. Frazer, B. Boashash, "Analytic signal generation—tips and traps", *IEEE Trans. Signal Process.* 42 (11) (1994) 3241-3245.
[7] B. Bouachache, "Representation temps-frequence", Tech. Rep. 373/78, Soc. Nat. ELF-Aquitaine, Pau, France, 1978, 56 pp.
[8] B. Bouachache, B. Escudie, J.M. Komatitsch, "Sur la possibilité d'utiliser la representation conjointe en temps et fréquence dans l'analyse des signaux modulés en fréquence emis en vibrosismiques", in: Seventh GRETSI Symp. on Signal Processing and its Applications, 1979, pp. 121/1-121/6.
[9] B. Boashash, A.P. Reilly, "Algorithms for time-frequency signal analysis", in: B. Boashash (Ed.), Time-Frequency Signal Analysis: Methods and Applications, Longman-Cheshire/Wiley, Melbourne/New York, 1992, pp. 163-181 (Chapter 7).
[10] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal—Part 2: Algorithms and applications", *Proc. IEEE* 80 (4) (1992) 540-568.
[11] J.M. O'Toole, B. Boashash, "Fast and memory-efficient algorithms for computing quadratic time-frequency distributions", *Appl. Comput. Harmo. Anal.* 35 (2) (2013) 350-358.